Gaussian Processes for Topology Inference of Directed Graphs

Chen Cui,* Paolo Banelli,† and Petar M. Djurić*
*Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY, USA
†Department of Engineering, University of Perugia, Perugia, Italy
Email:*{chen.cui, petar.djuric}@stonybrook.edu, †paolo.banelli@unipg.it

***Abstract*—In machine learning applications, the data are often high-dimensional and intrinsically related. It is often of interest finding the underlying structure and the causal relationships of the data and representing the findings with directed graphs. In this paper, we study multivariate time series, where each series is associated with a node of a graph, and where the objective is estimating the topology of the graph that reflects how the nodes of the graph affect each other, if at all. We propose a novel Bayesian method which allows for nonlinear and multiple lag relationships among the time series. The method is based on Gaussian processes, and it treats the entries of the adjacency matrix as hyperparameters. The method employs an automatic relevance determination (ARD) kernel and allows for learning of the mapping function from selected past data to current data. The resulting adjacency matrix provides the intrinsic structure and answers questions related to causality. Numerical tests show that the proposed method has comparable or better performance than state-of-the-art methods.**

***Index Terms*—topology inference, Gaussian processes, causality, ARD kernel**

I. Introduction

In many science and engineering problems, it is important to determine the underlying structure of observed data/signals because structures provide important insights about the system where the data come from. Given the significance of the problem, it is not surprising that learning the underlying structure of multidimensional data has been well studied in many fields, including biology [13], social sciences [19], and finance [10]. Examples include searching the functional connectivity within different areas in the brain, analysing relationships between individuals on social platforms and even entire societies, and studying interdependence of financial entities [20].

To find out the topology of a graph, a common method is to estimate the Laplacian matrix or the adjacency matrix of the graph. The work in [6] and [7] is about learning the structure of the data using the Laplacian matrix and with probabilistic interpretation of estimating a Gaussian Markov Random Field model. We observe that the use of the Laplacian entails that the estimated graph has a symmetric adjacency matrix, i.e., it is an undirected graph. While this approach may favor a low estimation workload, it is limited when it comes to dealing with directed single-way dependencies on the data, or when the generative model is nonlinear, as assumed in this paper. In [16], [17], the adjacency matrix can be seen as a graph filter, and in [1], [14] the inferred network topology is based on a structural equation model (SEM). The authors of [23], [25] proposed a kernel-based vector autoregressive (VAR) model to capture the nonlinearity and the dynamics of a graph. At the same time, the adjacency matrix can also be viewed as related to Granger causality [4], motivating the interpretation of the network of structural data-dependencies as an evidence of a causal relationships.

In this paper, we propose a novel Bayesian method to infer the directed adjacency matrix of a graph from time-series data observed at the nodes of the graph. At the same time, the objective is to estimate the causalities that exist in the network. We assume nonlinear functional relationships among the signals on the graph, where the functions, too, need to be estimated. Specifically, we propose to employ Gaussian processes (GPs) as a tool for learning the unknown mappings. The arguments of these functions are not only past local data but also past data from other nodes. The GPs are based on a predefined kernel, where the hyperparameters of the kernel encode the relevance determination of the arguments and thus, allow for automatic relevance determination (ARD). In this way, we can readily obtain the topology of the graph. Our method enables learning the strength of influence from all the selected previous data simultaneously, which means that the method is not dependent on the knowledge of specific time delays. We use GPs because they are data-efficient and flexible, and they can be viewed as a general tool for estimating nonlinear functions.

The ARD kernel has been successfully used in machine learning since it was first formulated in the framework of neural networks [3]. For example, [9] and [24] used ARD kernels to do feature selection for an SVM. Aside from feature selection, [8] proposed a method to infer the causality of two time series using the hyperparameter of Gaussian processes, whose kernel is an ARD kernel. Further, the idea of making inference based on the hyperparameters of GPs has been successfully applied in many ways. For instance, [22] introduced an online method for detecting change points, while [15] proposed a time-varying hyperparameter model to estimate time-varying functions.

The rest of the paper is organized as follows: In Section II, we give a brief overview of graphs, GPs and the ARD kernel. Then in Sections III and IV, we describe our model.

The authors thank the support of NSF under Award 2021002.

In Sections V and VI, we present numerical tests on different cases and provide concluding remarks, respectively.

II. Background

A. Graph and Graph Signal

Consider a graph denoted by $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{A})$ where $\mathcal{V}$ is a set of N nodes, $\mathcal{E}$ is a set of edges, and $\mathbf{A}$ is the graph's adjacency matrix. One way to describe the topology of a graph $\mathcal{G}$ is through $\mathbf{A}$, which can be symmetric or asymmetric and thus, implying if the graph is undirected or directed, respectively. The (n, m)th entry of $\mathbf{A}$ is $a_{nm} \in \{0, 1\}$. For an undirected graph, $a_{nm} = a_{mn}$, and if its value is equal to one, there is an edge *between* node m and node n. Similarly, for a directed graph, if a_{nm} is equal to one, there is an edge *pointing from* node m *to* node n. Similarly, with a weighted adjacency matrix $\mathbf{W}$ with positive entries $w_{n,m} \in \mathbb{R}$, we can represent the strength of coupling of the nth and mth nodes.

On a given graph, its signals are defined as follows. Consider an unordered set of data $\mathcal{S} = \{s_{\alpha_0}, ..., s_{\alpha_N}\}$, which are associated with $\mathcal{G}$. We assume each data in $\mathcal{S}$ are assigned to a single specific node in $\mathcal{G}$. Then the data $\mathcal{S}$ are ordered by the nodes in $\mathcal{G}$ and are given by an N-tuple $\mathbf{s} = \{s_1, ...s_n, ..., s_N\}$. We can think of $\mathbf{s}$ as a graph signal over $\mathcal{G}$ [5]. The nth element s_n in $\mathbf{s}$ is indexed by the node n of $\mathcal{G}$.

B. Gaussian Processes

Gaussian processes are a class of stochastic processes, which are used in machine learning for modeling functions [21]. More specifically, let $(\mathbf{x}_n, y_n)$, $n = 1, 2, \ldots, N$, be N input-output values, $\mathbf{y} = [y_1 \ y_2 \ldots y_N]^\top$, and $\mathbf{y} = \mathbf{f}(\mathbf{X})$, with $\mathbf{f} \in \mathbb{R}^{N\times 1}$ and $\mathbf{X} \in \mathbb{R}^{N\times d_x}$ being a matrix whose rows represent the inputs to the function $\mathbf{f}$, that is,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix}, \qquad \mathbf{y} = \mathbf{f}(\mathbf{X}) = \begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_N) \end{bmatrix}. \tag{1}$$

The idea behind GPs is to assume that the function's samples are jointly drawn from a Gaussian distribution instead of being deterministic. Mathematically, we have $\mathbf{f} \sim \mathcal{GP}(\mathbf{m}(\mathbf{X}), \mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X}))$, where $\mathbf{m}(\mathbf{X})$ is the mean function, $\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})$ is the covariance (kernel) function of the process, and $\boldsymbol{\theta}$ is the vector of hyperparameters of the GP i.e.,

$$\begin{aligned} \mathbf{m}(\mathbf{X}) &= \mathbb{E}[\mathbf{f}(\mathbf{X})], \\ [\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})]_{ij} &= \mathbb{E}[(f(\mathbf{x}_i) - m(\mathbf{x}_i))(f(\mathbf{x}_j) - m(\mathbf{x}_j))]. \end{aligned} \tag{2}$$

In practice, without loss of generality, we let the mean function to be $\mathbf{0}$, and by definition, the kernel must be positive definite [21].

C. Automatic Relevance Determination Kernel

A commonly used kernel for GPs is the squared exponential (SE) kernel with the following form:

$$k_l(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j)}{2l}\right), \tag{3}$$

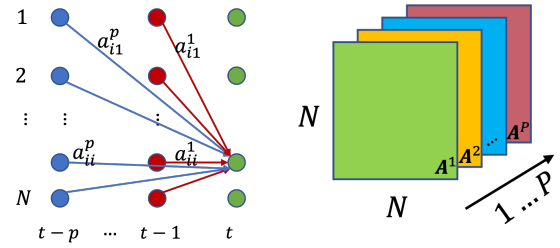
where l is a hyperparameter of the GP, also called a characteristic *length−scale*. The symbol l reflects the relationship between the distance one moves in the input space and how the function value changes in the output space [21]. Informally, if l is very small, the output is very sensitive to the change of the input, but if l is very large, small changes of the input do not affect the output much.

The ARD kernel is an extension of the SE kernel with the following form:

$$k_{ARD}(\mathbf{x}, \mathbf{x}') = \exp\left(-\sum_{i=1}^{d_x} \frac{(x_i - x_i')^2}{2l_i}\right), \tag{4}$$

where x_i is the ith entry of $\mathbf{x}$. Different from the SE kernel, for each component of the input vector, the ARD kernel assigns a different length−scale. If we think the input $\mathbf{x}$ is a vector of features, we can use the length−scale in deciding which features to discard from the input (the ones with small contributions or no contributions) and which not. In the sequel, we exploit the ARD kernel to infer (causal) relationships among the observed data.

III. Model Description



(a) Illustration of the model, where the arrows' directions suggest directions of influence. Different colors represent different time lags.

(b) Illustration of the dimension of the adjacency matrix $\mathbf{A}$, where each layer represents the adjacency matrix for a specific lag.

Fig. 1. Model Description

Assume $\mathbf{y}(t) \in \mathbb{R}^N$ is a column vector composed of measurements at time t on $\mathcal{G}$ with N nodes, whereas $y_n(t)$ denotes the graph signal of node n at t. Further, we assume that $y_n(t)$ is a function of the previous data on all (or some of) the nodes of $\mathcal{G}$. Specifically, we consider the data model

$$\mathbf{y}(t) = [f_1(\mathbf{Y}^P(t)), \ldots, f_N(\mathbf{Y}^P(t))]^\top + \mathbf{v}(t), \tag{5}$$

where $\mathbf{Y}^P(t) := [\mathbf{y}(t-P)^\top, \ldots, \mathbf{y}(t-p)^\top, \ldots, \mathbf{y}(t-1)^\top] \in \mathbb{R}^{1\times NP}$, with P being the discrete delay time span and $\mathbf{y}(t-p)$, the graph signals at time $t-p$. The model noise is $\mathbf{v}(t) \sim \mathcal{N}(\mathbf{0}, \sigma_v^2\mathbf{I})$, and f_n is the GP function of node n, as also visualized in Fig. 1(a). We also assume independent

modeling of past dependencies of the GP at each node, and thus, for the nth element of $\mathbf{y}(t)$ in (5), we write

$$y_n(t) = f_n(\mathbf{Y}^P(t)) + v(t). \tag{6}$$

For the signal at the nth node, we have

$$\widetilde{\mathbf{y}}_n = \mathbf{f}_n(\mathbf{X}) + \mathbf{v}, \tag{7}$$

where $\widetilde{\mathbf{y}}_n := [y_n(P+1), \ldots, y_n(T)]^\top \in \mathbb{R}^{(T-P)\times 1}$, $\mathbf{X} := [\mathbf{Y}^P(P+1)^\top, \ldots, \mathbf{Y}^P(T)^\top]^\top$, where $\mathbf{X} \in \mathbb{R}^{(T-P)\times NP}$, and the function $\mathbf{f}_n : \mathbf{X} \to \widetilde{\mathbf{y}}_n$ is drawn from a GP, i.e., $\mathbf{f}_n \sim \mathcal{GP}(\mathbf{0}, \mathbf{K}_n)$, where $\mathbf{K}_n \in \mathbb{R}^{T-P\times T-P}$ is the kernel function of the GP and has the following form:

$$\mathbf{K}_n = \sigma_n^2 \begin{bmatrix} k_{1,1}^n & k_{1,2}^n & \cdots & k_{1,T-P}^n \\ & & \vdots & \\ k_{T-P,1}^n & k_{T-P,2}^n & \cdots & k_{T-P,T-P}^n \end{bmatrix}, \tag{8}$$

where σ_n^2 is a hyperparameter, and $k_{i,j}^n$ is defined by

$$k_{i,j}^n = \exp\left(-\sum_{p=1}^{P} \sum_{m=1}^{N} \frac{(y_m(i+p-1) - y_m(j+p-1))^2}{2 l_{nm}^p} \right). \tag{9}$$

with the l_{nm}^p also being a hyperparameter.

IV. Proposed solution

In inferring the topology of the directed graph, we propose to use the kernel defined in (9). With (9), as anticipated by (7), we model the inputs of a specific node n by the *past* P values of the graph signals coming from all the graph nodes, including n itself.

If l_{nm}^p is small, a small change on the past history of node m will cause a large change to node n, for a specific delay index p, and vice-versa for large l_{nm}^p. Then, the set $\boldsymbol{\theta}_n := \{l_{nm}^p\}_{m=1,\ldots,N}^{p=1,\ldots,P}$ of hyperparameters can indicate which nodes contribute to the evolution of the *networked* signals, and which are not involved, i.e., which edges exist in the network and which do not. An easy way to think about the model is seeing each delay as a different layer of an adjacency matrix organized as a tensor, as shown in Fig. 1(b).

Equation (7) is our model, and based on the assumptions of the model, for the marginal likelihood of $\boldsymbol{\theta}_n$ (marginalized over the function $\mathbf{f}_n$), we can write

$$p(\widetilde{\mathbf{y}}_n|\mathbf{X}, \boldsymbol{\theta}_n) = \int p(\widetilde{\mathbf{y}}_n|\mathbf{f}_n, \mathbf{X}, \boldsymbol{\theta}_n) p(\mathbf{f}_n|\mathbf{X}, \boldsymbol{\theta}_n) d\mathbf{f}_n. \tag{10}$$

In practice, to train a GP model, the common method is to maximize the marginal likelihood in (10) instead of the posterior probability, since the integration $\int p(\widetilde{\mathbf{y}}_n|\mathbf{X}, \boldsymbol{\theta}_n) p(\boldsymbol{\theta}_n) d\boldsymbol{\theta}_n$ is usually intractable.

Thus, to learn the optimal set of hyperparameters $\boldsymbol{\theta}_n^*$, we maximize (10) using a gradient based approach on the marginal log-likelihood [21], which is expressed by

$$\log p(\widetilde{\mathbf{y}}_n|\mathbf{X}, \boldsymbol{\theta}_n) = -\frac{1}{2}\widetilde{\mathbf{y}}_n^\top \mathbf{K}^{-1} \widetilde{\mathbf{y}}_n - \frac{1}{2}\log|\mathbf{K}| - \frac{T-P}{2}\log 2\pi, \tag{11}$$

where $\mathbf{K} = \mathbf{K}_n + \sigma_v^2 \mathbf{I}$. The partial derivative of the marginal loglikelihood can be written as [21]

$$\frac{\partial}{\partial \theta_{(i)}} \log p(\widetilde{\mathbf{y}}_n|\mathbf{X}, \boldsymbol{\theta}_n) = \frac{1}{2}\widetilde{\mathbf{y}}_n^\top \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta_{(i)}} \mathbf{K}^{-1} \widetilde{\mathbf{y}}_n - \frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta_{(i)}}) = \frac{1}{2}\mathrm{tr}\left((\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - \mathbf{K}^{-1}) \frac{\partial \mathbf{K}}{\partial \theta_{(i)}} \right), \tag{12}$$

where $\boldsymbol{\alpha} = \mathbf{K}^{-1}\widetilde{\mathbf{y}}_n$ and $\theta_{(i)}$ is the ith element of $\boldsymbol{\theta}_n$. Thus, after we get the optimal set $\{l_{nm}^p\}^*$ from the minimization of (10), we use a threshold ϵ to determine the effective network, i.e., the nonweighted adjacency matrix, according to the following criterion:

$$a_{nm}^p = \begin{cases} 0, & \text{if } \frac{1}{l_{nm}^p} < \epsilon \\ 1, & \text{otherwise} \end{cases}, \tag{13}$$

and $w_{nm}^p = \frac{1}{l_{nm}^p}$ are the entries of the weighted and not-thresholded adjacency matrix $\mathbf{W}$.

V. Numerical Results

We analyzed the performance of the proposed method on three non-linear dynamic systems. Each system evolved according to specific relationships among the signals associated with the nodes of a network, and we compared the ability of our method to unveil the true network topology with that of other approaches. We note that our proposed method does not make any use of the generative signal and system model. The first example is a small dynamical system [18], whereas the second is a large one, i.e., a Lorenz 96 model [11]. Finally, the third example illustrates the applicability of the proposed method on a multiple delay test, detecting edges of a multi-layer causal model.

A. A Discretized Lorenz Attractor

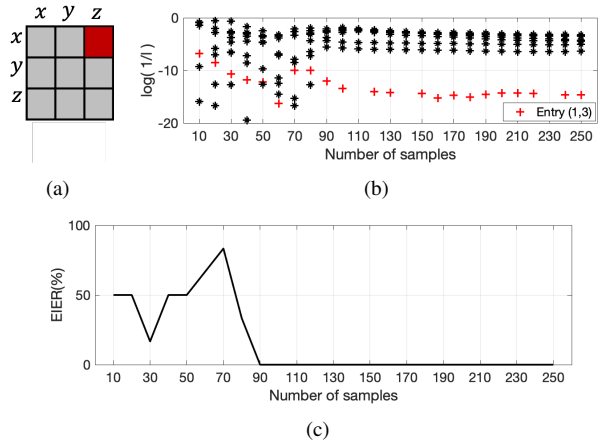
We considered the three-node network with one-lag memory associated with the discretized version of the Lorenz attractor [2], [12], described by

$$\begin{bmatrix} x^{t+1} \\ y^{t+1} \\ z^{t+1} \end{bmatrix} = \begin{bmatrix} x^t \\ y^t \\ z^t \end{bmatrix} + 0.01 \begin{bmatrix} 10(y^t - x^t) \\ x^t(28 - z^t) - y^t \\ x^t y^t - \frac{8}{3} z^t \end{bmatrix}, \quad t \geq 0. \tag{14}$$

Learning this network has been previously addressed in [18] by a kernel-based algorithm, specifically proposed to deal with the non-linear generative model of the data.

We point out that for benchmarking performance, the weighted adjacency matrix $\mathbf{W}$ is more informative than the binary connectivity matrix $\mathbf{A}$ and the associated heatmap. Thus, in Fig.2(b) we plotted the values of all the entries $\log(w_{nm}^p) = -\log(l_{nm}^p)$, for different lengths of the time series (starting with a series that is 10 samples long and increasing their sizes to 250 samples in steps of 10 samples). The black star marks in the figure are the entries that correspond to existing edges of the graph representing the model in (14), with the red plus mark being the only one that quantifies a missing edge. Figure 2(b), clearly shows that

the non-causal entry is well separated from the other ones. Figure 2(c) displays the evolution of the edge identification error rate $\mathbf{EIER} = \frac{|\mathbf{A}-\hat{\mathbf{A}}|_0}{N(N-1)}$ as a function of the number of samples, T, which has also been used in [18]. We set $P = 1$ in our model, i.e., exploiting knowledge of the one-lag coupling of the system, as in [18]. We point out that in our simulations we used the same initial conditions as in [18]. The method from [18] was capable to effectively detect all the edges after 200 samples, and ours, as seen from Fig. 2 after only 90 samples. With other initializations, our method had similar performance as the one presented in Fig. 2.



(a) (b)

(c)

Fig. 2. Discretized Lorenz Attractor. (a) The true adjacency matrix. (b) $\log(1/l_{mn})$ for all the entries of the adjacency matrix $\mathbf{W}$ with different number of samples, where $m, n = 1, 2, 3$. The red marks denote the entry $(1, 3)$, which corresponds to a non-existing edge, and the black marks to the remaining entries. The values of $(1, 3)$ for time series with lengths 110, 140, 230 samples are not plotted because their values are much smaller than -20. (c) The edge identification error rate with different number of samples when the logarithm of the threshold in equation (13) is equal to -8.

B. Lorenz 96 Model

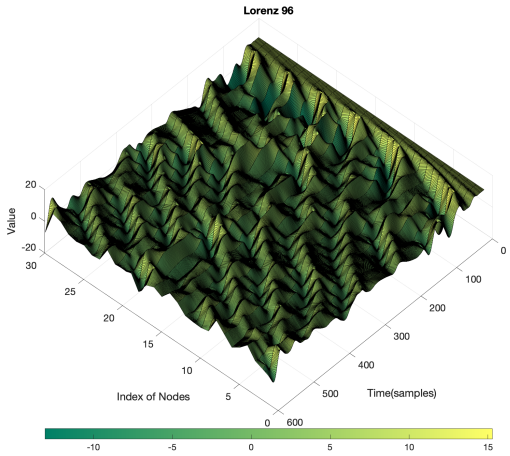
The Lorenz 96 model is defined as

$$\frac{dx_i}{dt} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F, \quad i = 1, 2, \ldots, N. \quad (15)$$

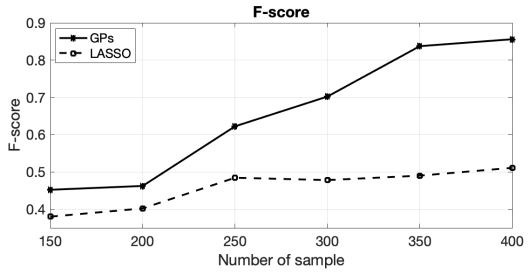
We generated the synthetic data shown in Fig. 3(a) by a numerical integration with a step size of 0.015, initializing all the nodes $\mathbf{x}(0) = F = 8$ and then adding a 0.01 perturbation to node 1. The lags in this case refer to the samples instead of time. We conducted a comparison with LASSO applied on a VAR model, and we measured the performance with the F-score, defined by

$$\text{F-score} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FN} + \text{FP})}, \quad (16)$$

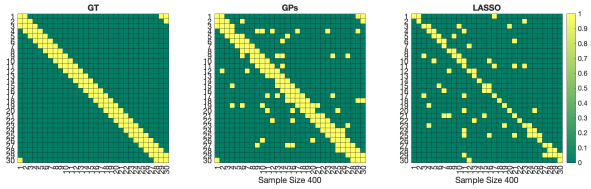
where TP stands for the number of true positives, FP for the false positives, and FN for the false negatives. The results are plotted in Fig. 3(b), which shows that the GPs can detect most of the true edges with fewer errors with respect to the VAR-LASSO, as pictorially highlighted also in Fig. 3(c).



(a) Illustration of a 30 dimensional Lorenz 96. The x axis is the index of the nodes, the y axis represents time, and the z axis is the value of the samples.



(b) Comparison of the F-scores obtained by the GPs and the LASSO method.



(c) Heatmap of the ground truth (presence/absence of edges), and heatmaps of the GP and the LASSO method with 400 samples. The yellow color corresponds to the presence of an edge.

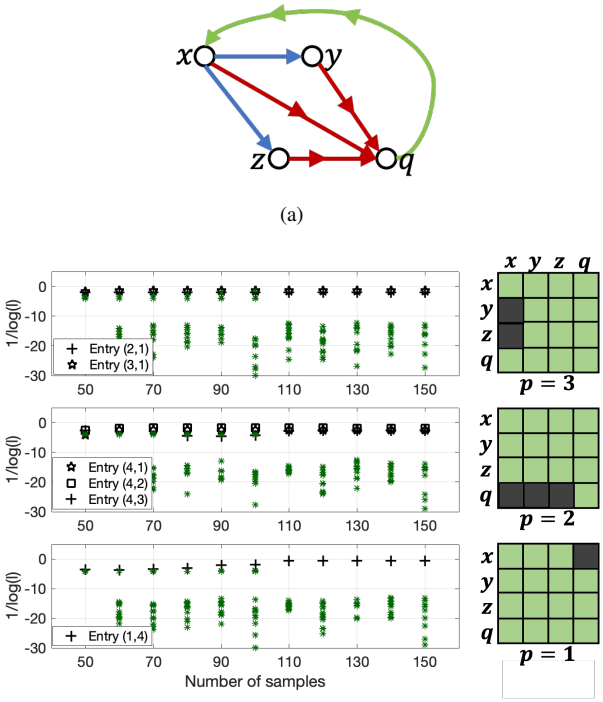
Fig. 3. Lorenz 96.

C. Multiple Delay

In the third experiment, we designed a small network to test the detection capabilities of the proposed approach in a multiple-delay problem. Each node interacted within the small network but with a different delay time. The data model is shown in Fig. 4(a), which graphically represents the dynamical system described by the following set of equations:

$$\begin{aligned} y(t) &= x(t-3)^2 + x(t-3), \\ z(t) &= \sin(x(t-3)) + 1, \\ q(t) &= x(t-2)z(t-2) + x(t-2)y(t-2) \\ &\quad + z(t-2)y(t-2) - 3, \\ x(t) &= \sin(q(t-1))/q(t-1). \end{aligned} \quad (17)$$

The initial values of x, y, z, q were drawn as i.i.d. samples from a Gaussian distribution, $\mathcal{N}(0, 0.5^2)$. Based on the above equation, we needed three adjacency matrices to represent the network topology, and they are plotted separately in Fig. 4(b). Figure 4(c) depicts the true adjacency matrix. The results clearly show that our GP-based method can identify the multiple delays by using 110 samples (or more).



(b) Value of $\log(1/l)$. (c) $\mathbf{A}$

Fig. 4. An example with multiple delays. (a) Data model description: the arrow directions reflect direction of causation. The different colors represent different delays, $p = 3$ (blue), $p = 2$ (red), and $p = 1$ (green). (b) The plots show all the elements of the estimated adjacency matrix; the entries which correspond to existing/non-existing edges are marked differently. (c) The true adjacency matrix for each layer. The top plots in (b) and (c) correspond to $p = 3$, the middle to $p = 2$ and the bottom to $p = 1$, respectively.

VI. Conclusion

In this paper, we proposed a GP-based method for estimation the topology of a graph from observed signals on the graph. The assumptions of the used model are mild, and the method can detect causation among the signals on the graph. We showed that its performance can be very good under different conditions, including non-linear dynamics, large systems, and even scenarios with multiple delays. Future directions of work include the following:

1) Principled ways of selecting thresholds for edge detection,
2) Extension of the proposed method to dynamic networks where the topology of the network varies with time,
3) Scalability of the method. Note that when the system delay P and the network size N increase, the number of parameters that need to be estimated is N^2P. If N and P are large, this becomes problematic.

References

[1] B. Baingana and G. B. Giannakis. Switched dynamic structural equation models for tracking social network topologies. In *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 682–686. IEEE, 2015.

[2] N. Biggs, N. L. Biggs, and B. Norman. *Algebraic graph theory*. Number 67. Cambridge university press, 1993.

[3] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[4] A. Bolstad, B. D. Van Veen, and R. Nowak. Causal network inference via group sparse regularization. *IEEE transactions on signal processing*, 59(6):2628–2641, 2011.

[5] P. Djuric and C. Richard. *Cooperative and graph signal processing: principles and applications*. Academic Press, 2018.

[6] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst. Learning laplacian matrix in smooth graph signal representations. *IEEE Transactions on Signal Processing*, 64(23):6160–6173, 2016.

[7] H. E. Egilmez, E. Pavez, and A. Ortega. Graph learning from data under laplacian and structural constraints. *IEEE Journal of Selected Topics in Signal Processing*, 11(6):825–841, 2017.

[8] G. Feng, J. G. Quirk, and P. M. Djurić. Inference about causality from cardiotocography signals using gaussian processes. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2852–2856. IEEE, 2019.

[9] C. Gold, A. Holub, and P. Sollich. Bayesian approach to feature selection and parameter tuning for support vector machine classifiers. *Neural Networks*, 18(5-6):693–701, 2005.

[10] M. Iloska, Y. El-Laham, and M. F. Bugallo. Graphical network and topology estimation for autoregressive models using gibbs sampling. *Signal Processing*, 190:108303, 2022.

[11] A. Karimi and M. R. Paul. Extensive chaos in the lorenz-96 model. *Chaos: An interdisciplinary journal of nonlinear science*, 20(4):043105, 2010.

[12] M. A. Kramer, E. D. Kolaczyk, and H. E. Kirsch. Emergent network topology at seizure onset in humans. *Epilepsy research*, 79(2-3):173–186, 2008.

[13] S. Lebre, J. Becq, F. Devaux, M. P. Stumpf, and G. Lelandais. Statistical inference of the time-varying structure of gene-regulation networks. *BMC systems biology*, 4(1):1–16, 2010.

[14] B. Liu, A. de La Fuente, and I. Hoeschele. Gene network inference via structural equation modeling in genetical genomics experiments. *Genetics*, 178(3):1763–1776, 2008.

[15] Y. Liu and P. M. Djurić. Gaussian process state-space models with time-varying parameters and inducing points. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1462–1466. IEEE, 2021.

[16] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro. Connecting the dots: Identifying network structure via graph signal processing. *IEEE Signal Processing Magazine*, 36(3):16–43, 2019.

[17] J. Mei and J. M. Moura. Signal processing on graphs: Causal modeling of unstructured data. *IEEE Transactions on Signal Processing*, 65(8):2077–2092, 2016.

[18] M. Moscu, R. Borsoi, and C. Richard. Online graph topology inference with kernels for brain connectivity estimation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1200–1204. IEEE, 2020.

[19] S. Myers and J. Leskovec. On the convexity of latent social network inference. *Advances in neural information processing systems*, 23, 2010.

[20] A. Nagurney and K. Ke. Financial networks with intermediation. *Quantitative Finance*, 1(4):441, 2001.

[21] C. E. Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[22] Y. Saatçi, R. D. Turner, and C. E. Rasmussen. Gaussian process change point models. In *ICML*, 2010.

[23] Y. Shen, B. Baingana, and G. B. Giannakis. Topology inference of directed graphs using nonlinear structural vector autoregressive models. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6513–6517. IEEE, 2017.

[24] T. Wang, H. Huang, S. Tian, and J. Xu. Feature selection for svm via optimization of kernel polarization with gaussian ard kernels. *Expert Systems with Applications*, 37(9):6663–6668, 2010.

[25] B. Zaman, L. M. L. Ramos, D. Romero, and B. Beferull-Lozano. Online topology identification from vector autoregressive time series. *IEEE Transactions on Signal Processing*, 69:210–225, 2020.